

**IN THE SPECIFICATION:**

Please replace paragraph [0005] with the following amended paragraph:

[0005] In addition to providing for user configuration of the browser's graphical user interface, users may also configure ~~configured~~ the operation of the browser. For example, a user may elect to disable or ~~[[our]]~~ enable the browser's ability to render Java content. Accordingly, browsers provide users with a high degree of flexibility in terms of the "look and feel" of the interface as well as the operation of the browser.

Please replace paragraph [0009] with the following amended paragraph:

[0009] One embodiment provides a method for controlling a first browser window using a second browser window. The method comprises opening a controlling browser window configured to control aspects of a controlled browser window and opening the controlled browser window which comprises a display area for rendering viewable content received from ~~received from~~ network locations. In one embodiment, the controlled browser window is opened from the controlling browser window. In another embodiment, the controlled browser window is opened before the controlling browser window.

Please replace paragraph [0025] with the following amended paragraph:

[0025] In some cases, embodiments of the invention may be implemented using specific programming languages and applications. For example, in one embodiment the JavaScript® scripting language is used to advantage with a Netscape® browser. However, it is understood that any reference to a particular embodiment is merely illustrative. Accordingly, embodiments may also be implemented using other browsers, such as Microsoft's Internet Explorer®, and other programming languages such as scripts including Visual Basic®, Perl and REX and even non-script languages including procedural languages (such as C) and object-oriented languages (such as Java and C++).

Please replace paragraph [0034] with the following amended paragraph:

[0034] The client computer 222 is generally under the control of an operating system 258, which is also located in memory 232. Illustrative operating systems which may be used to advantage include IBM's AIX® operating system, Linux® and Microsoft Windows® operating systems. More generally, any operating system supporting browser functions may be used. In one embodiment, the operating system 258 includes a timer 260. The timer 260 may be, for example, any one second timer suitable for monitoring the controlled window opened by the browser program 240 and for re-establishing the event handlers 252.

Please replace paragraph [0043] with the following amended paragraph:

[0043] Each AREA tag [[as]] has a corresponding HTML IMG tag. The IMG tag is linked to the AREA tag via the MAP tag. The event handlers for the IMG tag are MouseUp (which updates the image) and MouseDown (which updates the image and processes the action). Illustrative code for images corresponding to the code of Table I is shown below in Table II.

Please replace paragraph [0046] with the following amended paragraph:

[0046] If step 606 is answered affirmatively (i.e., the browser program 240 provides the necessary support), the method 600 proceeds to step 610 where the browser control program 250 sets up and enables the event handlers 252 for the controlling window. Processing ~~and~~ then proceeds to step 612 where the geometry and chrome for the controlled window of the browser program 240 is set up.

Please replace paragraph [0047] with the following amended paragraph:

[0047] At step 614, the method 600 queries whether the browser control program 250 is configured to implement a custom button bar (such as the button bar 402 shown in FIG. 4). If so, the geometry for the controlling window of the browser control program 250 is set up at step 616. At step 618, the custom button bar is set up within the controlling window. If, however, step 614 is answered negatively, processing proceeds to step 620

where the controlling window is set up as a hidden window. In another embodiment, the controlling window is set up as a minimized window, having an iconic representation on a task bar, for example. The particular manner in which the controlling window is implemented at step 620 is generally dependent upon the operating system 258 and/or a windows manager (if one exists for the computing environment of the client computer 222). In any case, processing ~~and~~ then proceeds to step 622.

Please replace paragraph [0049] with the following amended paragraph:

[0049] At step 626, the method 600 queries whether the browser control program 250 is configured to check for user activity. For example, it may be desirable to determine whether interaction between a user and the client computer 222 has occurred within a predetermined time period. This can be done by monitoring for the activity and registering instances of the activity. If no interaction has occurred ~~with-in~~ within the predetermined time period, the browser control program 250 may cause the browser program 240 to take some action, such as proceeding to a "home page", for example. Accordingly, if step 626 is answered affirmatively, a timer is set and started at step 628. The timer indicates a frequency with which a flag (or some variable state which indicated the user activity) is checked. The frequency with which user activity is checked may be varied according to application. Processing then proceeds to step 630. Processing also proceeds to step 630 from step 626 if step 626 is answered negatively.

Please replace paragraph [0049] with the following amended paragraph:

[0051] During a browsing session, the event handlers 243 and 252 serve to monitor ~~[[to]]~~ user input and produce a response according to the type of event handler. However, in one embodiment, the event handlers 243 are unloaded/disabled for each change in domain (i.e., each time the browser program 240 accesses a different URL). The event handlers 243 are unloaded as a result of the well-known "unload" event which occurs with a change in the URL. Accordingly, it is necessary to re-establish the event handlers 243 for the controlled window for each change in domain in order to continue monitoring user events. One embodiment for receiving and processing the unload

event (by one of the event handlers 243) is illustrated by method 700 described with reference to FIG. 7.

[0043] At step 814, the method 800 queries whether the user selected (i.e., clicked on) a link that will cause a new window to open. This determination can be made, for example, by recognizing that the URL contained in the address field includes "open", "[(')]", "http:", or "JavaScript:". If the query 814 is answered affirmatively, processing proceeds to step 816 where the request to open another window is ignored and the content located at the specified URL is rendered in the already open controlled window. Processing then proceeds to step 818.

Please replace the abstract with the following amended abstract. A replacement abstract is submitted on a separate sheet attached to this amendment.

Method, article of manufacture and apparatus for controlling the operation and appearance of browsers. A controlled browser window comprises a display area for rendering viewable content received from ~~received from~~ network locations. A controlling browser window is configured to control aspects of the controlled browser window. In one embodiment, a program implementing the controlling browser window comprises event handlers which produce a predetermined result in response to events occurring with respect to the controlled browser window. The event handlers may be re-established for each domain change of the controlled browser window.